Automatic Translation of Heterogeneous Data Models

Sheneela Naz, Yasir Majeed, and Amir Qayyum

***Abstract*—Schema translation is the process of transforming schemas in heterogeneous data models into a single target data model. Today there exist various data models to save useful data. The need of schema translation rises due to the existence of wide range of prominent, heterogeneous database applications. Due to the increasing requirement of interoperability of autonomous databases, multi database system which consists of homogenous or heterogeneous databases are required to work in a cooperative environment. In heterogeneous multi database system, the translation of one data model to another data model is critical. In this paper, we present a data model translation between heterogeneous models. Our approach translates the schema and data of one model to the target schema and data. We use the concept of metadictionary to achieve translation. We use XML as the standard to perform core translation. Our approach can adopt new upcoming models and can change them to old models easily.**

***Index Terms*—Data models, data translation, meta model, heterogeneous model.**

I. Introduction

Schema translation is the process to transform one data model to another data model. Schema translation is used to manage the heterogeneous data and to overcome heterogeneous databases interoperate ability problem. Data base schemas are represented in any data model such as entity-relationship model, object-oriented model, XML data model and a graph. Schema translation is to convert the component database schema (i.e. represented in any data model) to another targeted data base schema. Examples of schema translation are object oriented to relational database or from xml to relational database etc.

For the management of heterogeneous data, operator described in [1] can be used to generate database wrappers that are object-oriented to relational or XML to object-oriented and default database schemas from any other representations. The schema translation process is, given source schema M which is represented in source model and target schema model TS, generate the schema TS’ expressed in target schema model which is corresponding to M [1].

There are some issues that require the schema migration from one model to another. Migration of data is a difficult process due to potential differences in different models. The problem is much apparent even is case of small variations. There are some other issues are also exist such as object oriented wrappers are also required for xml data and translation from nested xml to relation database [1]. To solve these issues there are number of translation data models which are represented. A data model is a tool that provides an interpretation for data describing real world situations. It consists of a set of concepts to describe structures and contents of data, or data definition language, a set of operations to access the data, or data manipulation language (DML). The most prominent models are:

A. Relational Model (RM)

Relational model is the fundamental and simple model. Relational model is presented in the form of table instead of graph o shapes. Relational data has its own specific constructs which are used to define them such as

- Entities
- Relationship of entities
- Attributes of entities

B. Entity-Relationship Model (ERM)

ERM is the graphical representation of logical conceptual data and shows the relationship between entities. ERM has its own specific constructs which are used to define them such as

- Tables
- Relationship of tables
- Columns of tables
- Foreign keys defined over tables Relationships

C. Object Oriented Model (OOM)

It has more expression-power than the relational model. OOM has its own specific constructs which are used to define them such as

- Classes
- References among classes(functions)
- Fields or properties of classes

D. XML Model (XMLM)

It is emerging as a data format of internet era, there is an increasing need to efficiently store and query XML data. XML has emerged as a de-facto standard for data format on the web. The use of XML as a common format for representing, exchanging, storing and accessing data pose many challenges to database systems. XMLM has its own specific constructs which are used to define them such as

- Namespaces
- Elements
- Attributes of elements
- Relationships of elements & sub elements

There are some examples of translation between ER, SGML model to object oriented data model. Fig. 1 represents the ER schema. It is translated into Object Oriented data model which is presented in Fig. 2 [1].

Manuscript received February 23, 2012; revised April 24, 2012.

The authors are with the Center of Research in Networks and Telecommunication (CoReNeT), Mohammad Ali Jinnah University, Islamabad, Pakistan (e-mail: shahneela.cs@gmail.com)

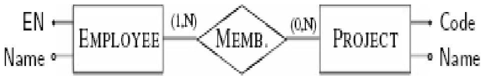

Fig. 1. ER schema

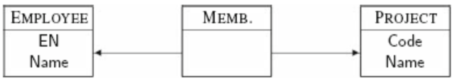

Fig. 2. Translation of ER schema into object oriented model

Some of the main issues in schema translation between two models are

- Incompleteness which refers to the potential loss of information which means that there might be some patterns in the source schema that can't be adequately represented in the target schema
- Eliminating redundancies caused by shared elements Performing optimizations

There are two schema Translation approaches [26] which are

- Direct Translation
- Indirect Translation

In direct translation, directly one data model is translated to another data mode. For this type of translation dedicated translation rules and algorithm are defined. It required a translation algorithm for each direction (e.g. from model A to model B and from model B to model A) as described in Figure 3. So this technique is model specific and there is not uniform approach for heterogeneous data model translation. The main disadvantaged of this approach is high complexity and low extensibility [26] because it required n*(n-1) translation algorithms for n number of data models. Low extensibility means, if new model is added in direct translation approach then 2*n algorithms are required for n number of data models.

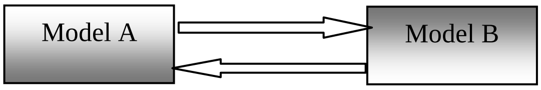

Fig. 3. Direct translation

Indirect translation is the less complex as compared to direct translation because it required only two steps for data translation [26]. During the process of translation from model A to model, it required intermediate model as described in Figure 4. In first step for data translation, model A is translated into intermediate model and then intermediate model data is converted into model B. The disadvantages of direct translation approach become the advantage of indirect translation. It required only 2*n algorithms for n number of data model translation. Another advantage is high extensibility, because it required only two algorithms for n number of data model translations.

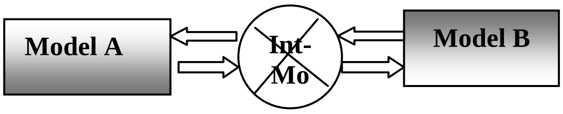

Fig. 4. Indirect translations

In this paper, we use the indirect approach for data model translations because it has the advantages as compared to the direct translation as described in above paragraphs.

II. Literature Survey

According to the literature, the problem of data model translation is drastically important with respect to model management. Model management is influenced by its two major advantages: low complexity and high extensibility. Metamorphosing from conceptual data models (e.ER, OR, Rel, XSD) to object oriented database model have been studied widely over the past years [16], [17], [18], [19].

Various proposals exist that consider schema translation. The term RETOO (RElational-ToObject-Qriented) is defined [20], which translates the relational schema into object oriented database schema using a set of translation rules. These translation rules derived from inclusion dependencies, key attributes and types of attributes. RETOO not only asserts the relational semantics but also enhance the object-oriented data modeling concepts using object-oriented data modeling concepts such as inheritance and aggregation. This translation scheme is simple and complete approach as compared to previous approaches [21], [22]. However this approach can be used just to convert relational to object oriented. This prototype is compared with other translation schemes [21], [23], [24].

"Schema-based translation system" [5] presented the TransScn translation system which is used the schema matching process for schema translation. This scheme is based on rule based method. Rules define the similarity and differences between two given schemas. The approach then uses rules to translate the instance of one schema into the instance of another schema. This system provides the convenient method for data conversion into another model. However if a new model needed to be translated into another model, we have to define new rules for translation.

In "automatic techniques for data model translation" [25] a framework for translation of data between heterogeneous models using a notion of meta-model is proposed. The approach takes a model, scheme and data as input; and translates into a target scheme and target data. Schemes are expressed in XML format in the approach. The output of the scheme is basically a set of metaprimitives like elements, relation, attribute, base domain, relationship, set, ordered list etc. In order to translate, the technique tried to translate each primitive in one domain to the associated primitive in the target model. The limitation of this approach is that it first defines the XML schema of every model's schema. So with the emergence of any new model we first need to define a mechanism to define its corresponding XML schema.

Some translation schemes or algorithms used the rules [26], [28] during the process of data model translation and some techniques used the dictionary [27] for translation. Paolo et al. [1] defined the generic translation scheme. It is mainly the implementation of ModelGen operator. This technique used both data dictionary and rules for data model translation. In super model, source data model elements are translated into generic homogenous data models. After that it applies the three main steps which are: first it removes the aggregation attributes and converts into one-to-many relationship and creates new abstracts. Second, it removes the many-to-many relationship and converts it into one-to-many aggregation and creates new abstracts. Third, finally it removes the one-to-many aggregation and creates the relationship between abstracts through the reference.

Paolo et al. [27] defined the generic translation method which is based on MIDST model. Midst model is already defined. In this translation method firstly import source model into data dictionary and stored it in metadata form. Translation step is performed according to the datalog program. Datalog program contains the basic transformation steps. Whereas in Paolo defined scheme which is runtime MIDST, it knows the source data model and its schema. In intermediate step, it transforms the source schema corresponding to the destination schema. Then it automatically generates the views based on datalog rules.

III. Proposed Technique

We present a technique that is model independent i.e. allows the definition of any possible model; and it can convert the schema of any model into any other model. Furthermore it also caters the emergence of new models i.e. it can facilitate new coming models. We use the concept of metadictionary somewhat like presented in Atzeni et al. [1]. However we used a simplified version of this dictionary as the dictionary just consist of meta constructs and there is no concept of super model in our approach.

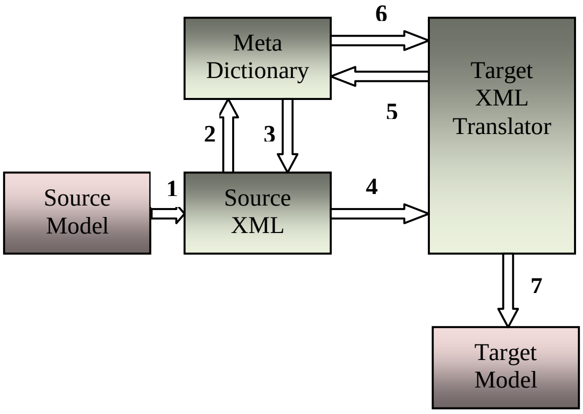

Fig. 5. Proposed architecture

According to Atzeni & Torlone et al. [29]; the constrcucts in the various models are rather similar. So they can be classified into a small number of categories called metaconstructs; and furthermore we can define translations on the basis of these metaconstructs. We use this concept as the basis of our technique. Our approach also uses an important observation by Hull and King’s [30] that the constructs used in most known models can be expressed by a limited set of generic metaconstructs. Fig. 5 shows our proposed technique.

Our proposed scheme takes a source model M_s as input and translates it into a target model M_t. It accomplishes this by converting the schema of source S_s into target schema S_t and it also converts the data of source model D_s into target data D_t. Both can be different models or even same ones. After taking the source model as input (step 1) the *Source XML Translator* consults the metadictionary to find the constructs of the input model (step 2). Our dictionary just describes the constructs of the models, properties of the constructs and describes the general description of the constructs as defined by the model. Some part of metadictionary is somewhat similar as the metamodel proposed in [25]. The contents of the metadictionary are defined by the domain (model) experts. They will define each of the constructs and their properties in the metadictionary. Unlike the metadictionary of [1] our metadictionary does not copy the source schema into itself but it identifies the constructs of the input model and returns them back to the source XML translator (step 3). Source XML Translator then generates the description of those constructs in the form of XML document. It also generates the XML file for the instances (data) of the input model and sends it to the Target XML Translator. Source XML translator passes both XML documents (Schema and data) as the input to the *Target XML Translator* (step 4). Target XML Translator will perform the actual translation of source schema into target schema. This will take the constructs from the source schema (step 5) and and maps them onto the corresponding constructs in the target schema (step 6). This translation is based on the assumption described earlier that the constructs in the various models are rather similar [29]. Basically this component will map the construct of the source model to the corresponding (similar) item. Based on the similarity assumption we can apply generic transformation as well as the transformation based on some rules in case of the different versions of the same model. Apart from generic transformation another way can be to define models (Decision tree or classification models) to map the constructs of one data model to the constructs of the other model. The output of the target XML translator is the target schema and target data in the form of XML (step 7). Finally the target schema and data is generated.

Our scheme makes use of the existing knowledge from the domain expert of the different models. The output schema will be a valid schema for the target model. The output of source XML Translator if saved can be used to translate the same schema of the model to the other various schemas of the model.

IV. Conclusion

We presented an approach for data model translations for heterogeneous models. We use metdictionary for the purpose of translation. Our scheme uses the XML as the core standard of translation, as XML is used as the standard for conversion purposes in various domains. We exploit the observation of similarity of constructs in various domains. We use generic transformation of models to avoid complexity of the process. Our approach has the ability to adopt new models and there is no need to add any module for new coming models.

References

[1] P. Atzeni, P. Cappellari, R. Torlone, P. A. Bernstein, and G, Gianforme, “Model-Independent Schema Translation,” *the VLDB Journal, Springer-Verlag*, 2008.

[2] J. Fonga, H. K. Wonga, and Z. Chengb, “Converting Relational Database into XML documents with DOM,” *Information and Software Technology, Elsevier*, vol. 45, pp. 335–355, 2003.

[3] T. Milo and S. Zohar, Using Schema Matching to Simplify Heterogeneous Data Translation, in *Proc. of the 24th VLDB Conference,* New York, USA, 1998.

[4] S. Monk, John A. Marianib, B. Elgalalb, and H. Campbell, “Migration from Relational to Object-oriented Databases,” *Information and Software Technology*, 1996.

[5] M. S. Zohar. “Using Schema Matching to Simplify Heterogeneous Data Translation,” in *Proc. of the 24th VLDB Conference*, New York, USA, 1998.

[6] P. Atzeni, P. Cappellari, R. Torlone, P. A. Bernstein, and G. Gianforme, “Model independent schema translation,” *VLDB J.* vol. 17, no. 6, pp. 1347-1370 , 2008.

[7] H. Robert Warren et al. *Multicolumn Substring Matching for Database Schema Translation*. The VLDB Journal, ACM 2006.

[8] A. P. Bernsteinet, S. Melnik, and P. Mork. “Interactive Schema Translation with Instance-Level Mappings,” The *VLDB Conference*, Norway, 2005.

[9] P. J. McBrien and A. Poulovassilis, “A Formal Framework for ER Schema Transformation,” in *Proc of ER’97*, Vol. 1331, pp. 408-421, 1997.

[10] J. Fong and S. K. Cheung, “Adding an Object-Oriented Interface to Relational Database Using Frame Model,” *International Journal of Information Technology*, Vol. 7, 1999.

[11] D. Lee, M. Mani, F. Chiu, and W. W. Chu, NeT and CoT: Translating *Relational Schemas to XML Schemas using Semantic Constraints*. CIKM’02, 2002.

[12] M. Kolp and E. Zimányi, “Enhanced ER to Relational Mapping and Interrelational Normalization,” *Information and Software Technology*, vol. 42, no. 15, pp. 1057-1073, 2000.

[13] R. A. Handa, J. R. Cordy, and T. P. Martin, “Schema translation using structural transformation,” in *Proc. of CASCON Toronto*, pp. 202-215, 1994.

[14] P. Martin, “Information Capacity Preserving Translations of Relational Schemas Using Structural Transformation,” Technical Report 95-392, Dept. of Computing and Information Science Queen’s University at Kingston, 1995.

[15] D. Lee, et al., “Constraint preserving transformation from XML Document type definition to relational schema,” *Int’l Conf. on Conceptual Modeling (ER)*, Salt Lake City, UT, 2000.

[16] C. Batini, S. Ceri, and S. Navathe, “Database Design with the Entity-Relationship Model. Benjamin and Cummings Publ. Co,” *Menlo Park*, CA .1992.

[17] M. Boyd and P. McBrien, “Comparing and transforming between data models via an intermediate hypergraph data model,” J. *Data Semantics IV*, pp. 69–109, 2005.

[18] J. L. Hainaut, “The transformational approach to database engineering,” In *GTTSE, LNCS*, vol. 4143, pp. 95–143. Springer, Berlin. 2006.

[19] P. Atzeni, G. Gianforme, and P. Cappellari, “Reasoning on data models in schema translation,” in*. FOIK Symposium*, *LNCS*, vol. 4932, pp. 158–177. Springer, Berlin, 2008.

[20] H. Ibrahim, S. L. Ki, A. Mamat, and Z. Muda, “Developing Translation Rules for Converting Relational to Object Oriented Database Conceptual Schema,” *Pertanika J. Sci. and Techno*, vol. 13, no. 1, 2005.

[21] CASTELLANOS, M. F. SALTOR and M. GARCiA-SOLACo, “Semantically enriching relational databases into an object oriented semantic mode*l*,” 1994.

[22] P. Mcbrjen and A. Poulovassius. “Automatic migration and wrapping of database applications-A schema transformation approach,” *Department of Computer Science Technical Report*, King's College London, 1998.

[23] Fong, *Converting relational to object-oriented databases*, Publication of SIGMOID Record, 26, o. I, 1997.

[24] P. Stanisic, “Database transformation from relational to object-oriented database and corresponding query translation,” in *Workshop on Computer Science and Information Technology CSIT*, pp. 199-208.

[25] P. Papotti and R. Torlone, “Automatic Techniques for Data Model Translation,” in *Proc. of the 31st VLDB Conference*, Trondheim, Norway, 2005.

[26] S. Y. W. Su, S. C. Fang, and H. Lam, “An object-oriented rule-based approach to data model and schema translation,” *Technical report TR-92-015*, University of Florida, Gainesville, Florida, 1992.

[27] P. Atzeni, L. Bellomarini, F. Bugiotti, and G. Gianforme, *A runtime approach to model independent schema and data translation*, *EDBT*, pp. 275-286.

[28] H. Ibrahim, S. L. Ki, A. Mamat, and Z, Muda, “Developing Translation Rules for Converting Relational to Object Oriented Database Conceptual Schema,” Pertanika J. Sci. and Techno. vol. 13, no.1. pp. I – 21, 2005.

[29] P. Atzeni and R. Torlone, “Management of Multiple Models in an Extensible Database Design Tool. In Fifth International Conference on Extending Database Technology (EDBT 96),” *Avignon, France, Lecture Notes in Computer Science*, Springer-Verlag, 1996

[30] R. Hull and R. King, Semantic database modelling: Survey, applications and research issues. ACM Computing Surveys, vol.19, no. 3, pp. 201–260, Sept. 1987.